

وزارة التعليم العالي والبحث العلمي

BADJI MOKHTAR- ANNABA UNIVERSITY
UNIVERSITE BADJI MOKHTAR ANNABA



جامعة باجي مختار- عنابة

Année : 2019

Faculté: Sciences de l'Ingénierat
Département: Electronique

MEMOIRE

Présenté en vue de l'obtention du diplôme de : LICENCE

Intitulé
Communication entre deux Arduinos via
Bluetooth

Domaine : Sciences et Technologie

Filière : Télécommunications

Par : BENSALÉM Rania
BENDIF Nour El Houda
KHEZAZNA Safa

DEVANT Le JURY

Directeur de mémoire : YAHY Amira

Examineur :

Sommaire

Résumé	3
Liste des figures	4
I.INTRODUCTION GENERALE.....	5
II.PROBLEMATIQUE.....	6

Partie théorique

III .MATERIEL.....	6
III.1Arduino.....	6
III.1.1 Définition d' Arduino.....	6
III.1.2 Les types d'Arduino	6
III .1.3Definition d'Arduino UNO	7
III.1.4 Le programmation avec l'IDE Arduino.....	7
III.2 Les modules Bluetooth HC-05 et HC-06	8
III.3 Autres matériels	8
III.3.1Une platine d'expérimentation.....	8
III.3.2 les fils	9
III.3.3 Câble USB	9
III.3.4 Une diode électroluminescente (LED)	9

Partie pratique

IV .METHODE.....	10
IV.1 L'organigramme	10
IV.2 Le montage Arduino-Bluetooth	11
IV.2.1 Montage Arduino Bluetooth HC 05 « Maitre »	11
IV.2.2 Montage Arduino Bluetooth HC 06 « Esclave ».....	11
IV.3 Configuration des modules Bluetooth	12
IV.3.1 La configuration du module HC-05	12
IV.3.2 La configuration du module HC-06	13
IV.4 Le montage entre deux Arduino-Bluetooth	15
IV.5 Partie programmation	15
IV.5.1 Le program Arduino 1 « Maitre ».....	15
IV.5.2 Le program Arduino 2 « Esclave »	16
V.CONCLUSION GENERALE.....	19

ملخص :

اختلفت استعمالات الاردوينو في شتى المجالات و من هنا اخترنا مجال الاتصال اللاسلكي المتبوع بالبلوتوث عبارة عن إرسال معلومات من الاردوينو الأول المزود بالبلوتوث المبرمج كباعث إلى الاردوينو الثاني الذي زود ببلوتوث مستقبل .

Résumé :

on utilise l'arduino dans plusieurs domaines .A partir de là nous avons choisi la communication entre deux arduinos par bluetooth , le premier arduino est équipé d'un bluetooth de transmission des informations appelé le maitre le deuxième bluetooth reçoit des informations appelé l'esclave.

Abstract :

There is difference uses of the arduino in many fields . From this field we choose the wireless bluetooth communication (sending information from the first arduino (master) to the second one (slave) that has receiving bluetooth.

La liste des figures :

- Figure 1 : Carte électronique
- Figure 2 : Environnement de programmation
- Figure 3 : La carte d'Arduino UNO
- Figure 4 : Interface de programmation avec l'IDE Arduino
- Figure 5 : Module HC-05
- Figure 6 : Module HC-06
- Figure 7 : Une platine d'expérimentation
- Figure 8 : Les fils
- Figure 9 : Le câble USB
- Figure 10 : Une diode électroluminescente (LED)
- Figure 11 : L'organigramme
- Figure 12 : Montage Arduino-Bluetooth HC 05
- Figure 13 : Montage Arduino-Bluetooth HC 06
- Figure 14 : Code du module HC-05
- Figure 15 : Le moniteur série
- Figure 16 : Moniteur série du module HC-05
- Figure 17 : Code du module HC-06
- Figure 18 : Moniteur série du module HC-06
- Figure 19 : Connecter le maître à l'esclave
- Figure 20 : Le montage entre deux arduino-Bluetooth

I . INTRODUCTION GENERALE :

La communication est un rapport d'interaction qui s'établit lorsque l'émetteur et le récepteur sont en contact. C'est à travers elle que la relation se constitue, se développe et évolue ; elle représente donc la dimension dynamique du lien. Il n'y a pas de relation sans communication.

La notion d'information n'est pas synonyme de communication : elle émerge de l'acte de communiquer. On la définit comme un élément transmis lors d'un échange de communication, ou plus simplement un message. Le transport de ce message s'effectue grâce aux moyens de communication (l'oral, l'écrit, le geste, l'image, ...) et aux supports (le papier, le téléphone, l'email, ...). La mise en forme renvoie aux genres de la communication.

Par ailleurs, la communication a de nombreuses dimensions (verbale et non verbale, communication de groupe, communication d'entreprise, etc.) et confère à ce terme une portée large. On peut ainsi distinguer quatre niveaux :

1. Les pratiques effectives de la communication
2. Les outils et techniques que l'on met en oeuvre
3. Les théories sous-jacentes
4. Les enjeux qui sont associés à la communication

Dans ce projet, nous essaierons de réaliser une communication entre deux cartes arduinos via Bluetooth, nous détaillerons les différentes étapes d'une telle réalisation.

II. PROBLEMATIQUE :

Avec l'arrivée de l'IOT (internet des objets) et le besoin de contrôle, les appareils doivent désormais faire plus que de remplir les fonctions de base pour lesquelles ils ont été conçus: ils doivent être capables de communiquer avec d'autres appareils, comme un téléphone portable, entre autres. Il existe différents systèmes de communication qui peuvent être adaptés pour la communication entre appareils. Parmi ces systèmes la communication par Bluetooth.

Dans le cadre de ce projet, il a été discuté de la manière de mettre en place un système simple de communication par Bluetooth entre deux destinations différentes pour envoyer et recevoir des informations entre elles via l'Arduino.

Partie théorique

III.MATERIEL

III.1 Arduino

III.1.1 Définition d' Arduino :

L'Arduino est une plate-forme de prototypage d'objets interactifs à usage créatif constituée d'une carte électronique et d'un environnement de programmation. Cet environnement matériel et logiciel permet à l'utilisateur de formuler ses projets par l'expérimentation directe avec l'aide de nombreuses ressources disponibles en ligne.



Figure 1



Figure 2

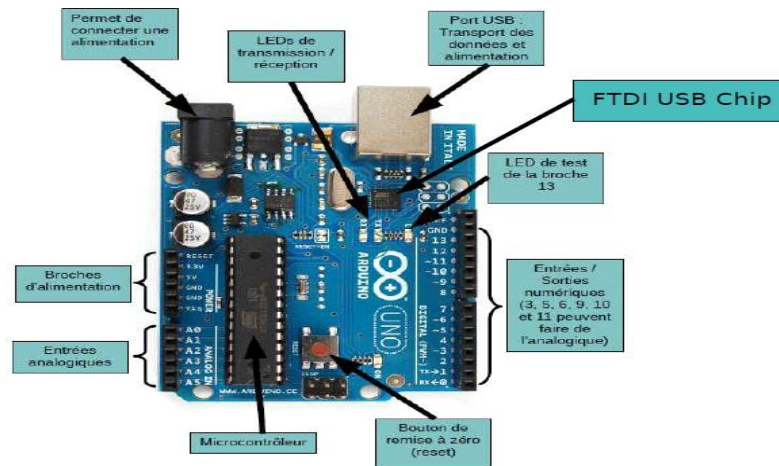
III.1.2 Les types d'Arduino :

- a- Arduino UNO
- b-Arduino MEGA
- c-Arduino NANO

III.1.3 Definition d'Arduino UNO :

Grâce à cette carte, on peut faire plusieurs dizaines de programmes. Pour cela, la carte dispose de 14 Entrées / Sorties et mesure 68,6 mm x 53,3 mm. Arduino uno est une multiplateforme qui peut fonctionner sous windows, macintosh et linux, avec un environnement de programmation clair et simple. De plus, c'est une carte qui n'est pas chère, en open source qui contient de nombreuses bibliothèques disponibles avec diverses fonctions implémentées. Dans notre réalisation, nous avons utilisé ce type de carte arduino.

Figure 3



III.1.4 La programmation avec l'IDE Arduino :

L'ARDUINO fournit un environnement de développement (IDE) avec un éditeur de source, les opérations de compilation et de chargement dans la mémoire du microcontrôleur étant ramenées à des clics sur des boutons dans l'IDE.

La communication entre le PC et la carte se fait via le port USB, moyennant installation d'un driver adapté (fourni par ARDUINO).

UN PROGRAMME ARDUINO = une fonction setup() + une fonction loop()

– la fonction setup() : doit contenir les initialisations (times, interrupts...)

– la fonction loop() : fonction répétée indéfiniment

La fonction setup() configure la broche PB5 (connexion n°13 sur la carte) en sortie, à l'aide de la fonction Arduino pinMode().

La fonction loop() décrit ensuite ce qui sera répété indéfiniment :

mettre PB5 à 1 pendant 200ms puis mettre PB5 à 0 pendant 1s, et ainsi de suite.

Figure 4



III.2 Les modules bluetooth HC-05 et HC-06 :

Il existe deux sortes de module bluetooth, tous deux compatibles arduino et utilisables sur un breadboard (plaque d'essai en français). On les distingue par le nombre de pattes d'entrées / sorties :

HC-05 : 6 sorties. Ce module peut être « maître » (il peut proposer à un autre élément bluetooth de s'appairer avec lui) ou « esclave » (il ne peut que recevoir des demandes d'appairage). Ce module fait l'objet d'un autre article car il y a quelques différences pour le régler.

HC-06 : 4 sorties. Ce module ne peut être qu'esclave. C'est ce module que nous utilisons dans ce projet.



Figure 5

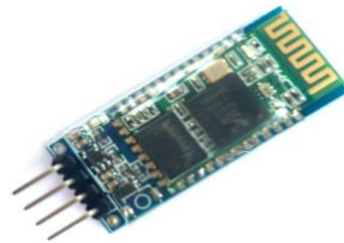


Figure 6

Les deux modules peuvent être utilisés en mode COMMANDE, pour les programmer avec des « commandes AT », ou en mode DATA, pour échanger des données.

III.3 Autres matériels :

III.3.1 Une platine d'expérimentation :

Une platine d'expérimentation ou platine de prototypage (appelée en anglais breadboard, solderless breadboard, protoboard, plugboard ou encore Labdec du nom de la marque la plus répandue) est un dispositif qui permet de réaliser le prototype d'un circuit électronique et de le tester. L'avantage de ce système est d'être totalement réutilisable, car il ne nécessite pas de soudure.

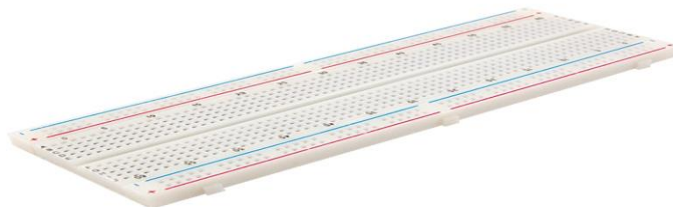


Figure 7

III.3.2 les fils :

Les fils électriques sont des composants servant à transmettre des signaux ou de l'énergie électrique.

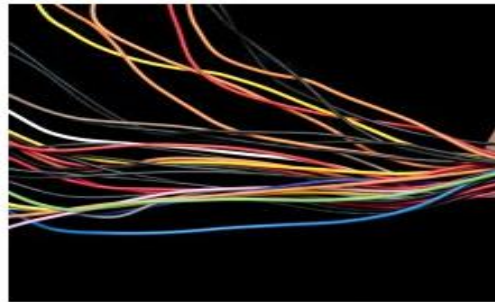


Figure 8

III.3.3 Cable USB :

Le câble USB est le câble le plus utilisé dans le monde , Les câbles USB sont utilisés pour communiquer entre arduino et le PC.



Figure 9

III.3.4 Une diode électroluminescente (LED) :

Il s'agit d'un dispositif optoélectronique capable d'émettre de la lumière lorsqu'il est parcouru par un courant électrique.

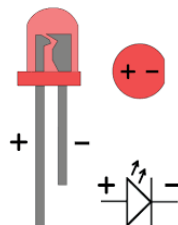


Figure 10

Partie pratique

IV .METHODE : Télécommande une LED

On utilise un Arduino pour contrôler une LED connectée à un second Arduino. La communication est à sens unique et il n'y a pas de vérification d'erreur. Arduino 1 (maître) envoie simplement les commandes LEDON et LEDOFF à l'Arduino 2 (esclave). Lorsque l'Arduino 2 reçoit les commandes, il configure le voyant en conséquence.

IV.1 L'organigramme :

Avant de passer à la programmation, nous devons réaliser un organigramme qui explique le déroulement des différentes séquences, tant intérieures qu'extérieures, il comportera plusieurs boucles dont la fin d'exécution succède toujours à son commencement.

La figure 11 représente l'organigramme qui explique le fonctionnement du système réalisé.

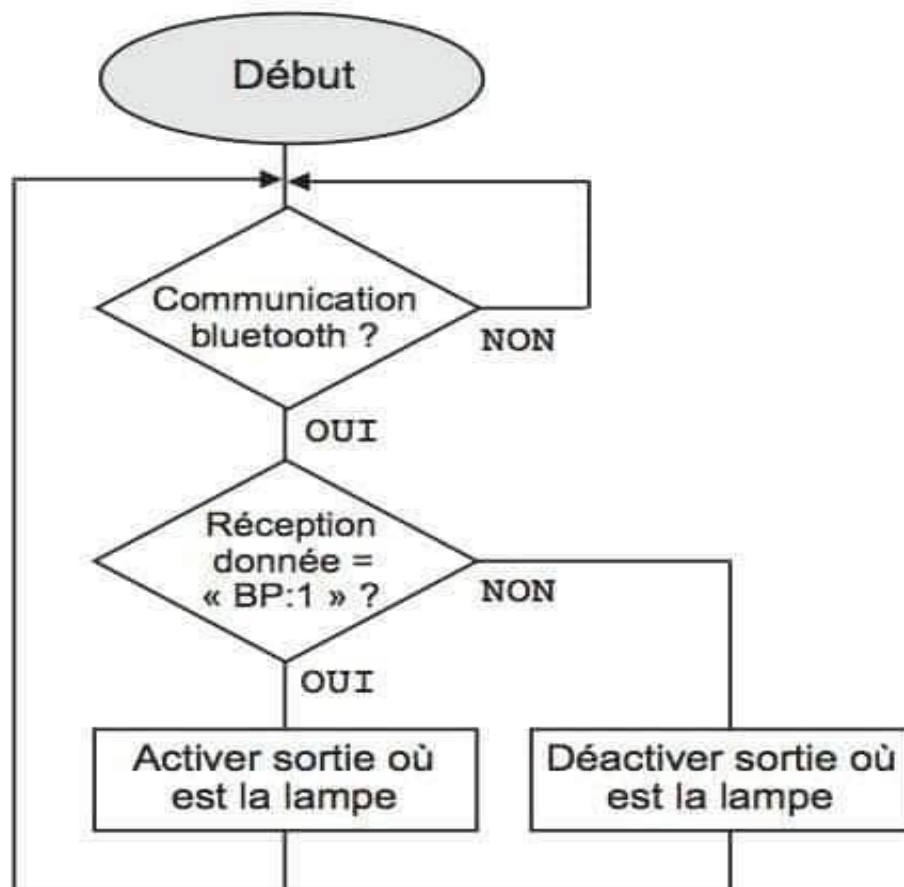


Figure 11

IV.2 Le montage Arduino-Bluetooth :

IV.2.1 Montage Arduino Bluetooth HC 05 « Maitre » :

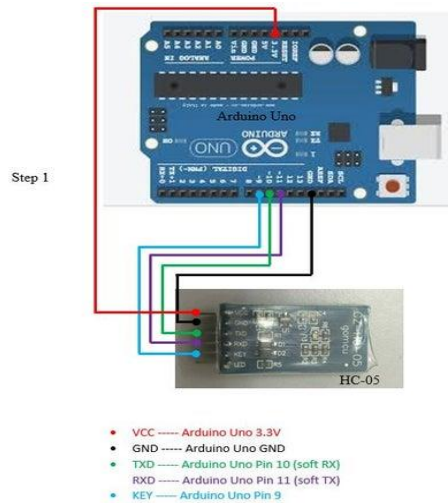


Figure 12

Tout d'abord on connecte les composants comme indiqué ci-dessus.

Ensuite, avant de connecter la carte Arduino au câble USB, ENLEVEZ LE FIL VCC DU HC-05. Ce n'est qu'après avoir retiré le fil VCC que vous connectez l'USB à la carte Arduino.

IV.2.2 Montage Arduino Bluetooth HC 06 « esclave » :

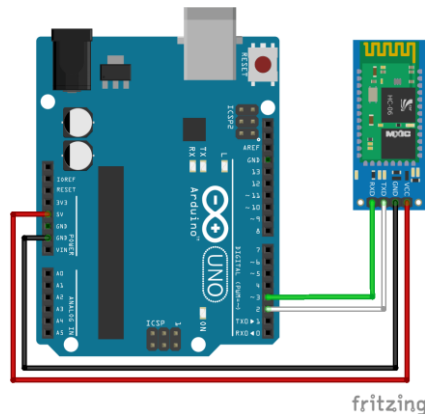


Figure 13

Il y a 4 pattes dont uniquement sont utilisées couramment. Dans l'ordre :

- VCC : Alimentation en 5V d'Arduino.
- GND : Relier au GND d'Arduino.
- RX : Broche de réception. Typiquement connecté à la broche de transmission (TX) de l'Arduino.
- TX : Broche de transmission. Typiquement connecté à la broche de réception (RX) de l'Arduino.

IV.3 Configuration des modules Bluetooth :

La configuration du module Bluetooth (HC-05 et HC-06) est essentielle pour vérifier le bon fonctionnement du module et pour modifier ses paramètres notamment lorsqu'on utilise plusieurs modules. Le module doit être alimenté mais non-apparu (la LED doit clignoter).

Le code programmation suivant permet de modifier le nom, le code PIN et la vitesse de communication du module HC-05 et HC-06.

IV .3.1 La configuration du module HC-05 :

On télécharge le code suivant sur l'Arduino à l'aide d'un câble USB.

```
HC-05 $
#include <SoftwareSerial.h>

SoftwareSerial BTSerial(10, 11); // RX | TX

void setup()
{
  Serial.begin(9600);
  pinMode(9, OUTPUT); // this pin will pull the HC-05 pin 34 (key pin) HIGH to switch module to AT mode
  digitalWrite(9, HIGH);
  Serial.println("Enter AT commands:");
  BTSerial.begin(38400); // HC-05 default speed in AT command mode
}

void loop()
{
  // Keep reading from HC-05 and send to Arduino Serial Monitor
  if (BTSerial.available())
    Serial.write(BTSerial.read());

  // Keep reading from Arduino Serial Monitor and send to HC-05
  if (Serial.available())
    BTSerial.write(Serial.read());
}
```

Figure 14

Après avoir installé le code, nous déconnectons le câble de connexion d'ordinateur Arduino. Maintenez enfoncé le petit bouton du HC-05 et connectez le câble Arduino à l'ordinateur. Maintenant, le voyant rouge sur le Bluetooth clignotera toutes les 3 secondes. Cela signifie que le module HC-05 est passé en mode de configuration.

On ouvre maintenant le moniteur série en appuyant sur Ctrl + Maj + M ou sur le menu Outils du programme Arduino.

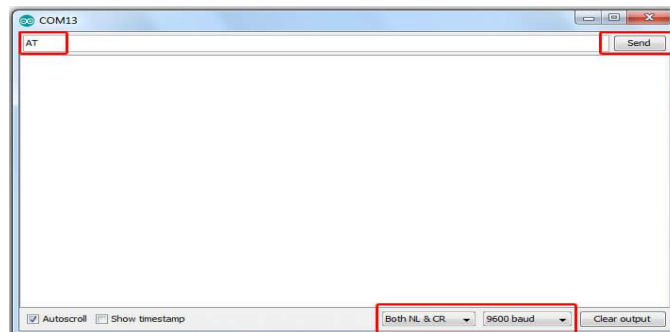


Figure 15

Après avoir sélectionné les champs ci-dessus, on ferme le moniteur série et on rouvre. Maintenant que nous avons tapé AT en lettres majuscules, le mode de configuration a été passé avec succès si OK est affiché.

Si les commandes saisies sont répondues par "OK", cela signifie que les paramètres mentionnés ont été personnalisés.

Les commandes AT du module HC-05 :

Les commandes AT permettent d'obtenir des informations sur le module Bluetooth ou de modifier les paramètres.

AT : Vérifier la connexion avec le module

AT+NAME : Voir le nom du module

AT+ADDR : Voir l'adresse du module

AT+VERSION : Connaitre la version

AT+UART : Connaitre la vitesse de connexion

AT+ROLE: Voir le rôle du module (1=master/ 0=slave/ 2=esclave boucle)

AT+RESET : Redémarrage du module et sortir du mode AT

AT+ORGL : Restaurer le module d'usine

AT+PSWD: Consulter le mot de passe

AT+BIND=adresse,du,slave (on a remplacé les : par des ,) , permet de connecter un module master à un slave

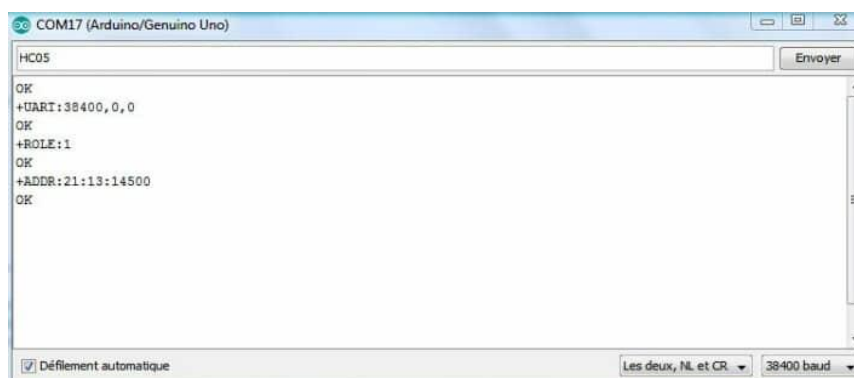


Figure 16

IV .3.2 La configuration du module HC-06 :

```
HC-06 $
#include <SoftwareSerial.h>

SoftwareSerial BTSerial(10, 11); // RX | TX

void setup()
{
  Serial.begin(9600);
  pinMode(9, OUTPUT);
  digitalWrite(9, HIGH);
  Serial.println("Enter AT commands:");
  BTSerial.begin(38400); // HC-06 default speed in AT command mode
}

void loop()
{
  // Keep reading from HC-06 and send to Arduino Serial Monitor
  if (BTSerial.available())
    Serial.write(BTSerial.read());

  // Keep reading from Arduino Serial Monitor and send to HC-06
  if (Serial.available())
    BTSerial.write(Serial.read());
}
```

Figure 17

- On ouvre maintenant le moniteur série en appuyant sur Ctrl + Maj + M ou sur le menu Outils du programme Arduino.
 - Pour tester la communication, tapez AT dans le monitor série de l'IDE Arduino. Assurez-vous de sélectionner le bon baudrate (9600) et « Les deux, NL et CR » dans les options de communication. Si tout va bien, le module doit répondre OK. Si ça ne fonctionne pas vérifiez le branchement et la version du module.
 - Pour modifier le nom du module, tapez AT+NAMEnom_module, le module devrait répondre OKsetname.
 - Pour modifier le code PIN du module, tapez AT+PINxxxx, le module devrait répondre OKsetPIN.
 - Pour modifier la vitesse de communication du module (seulement si nécessaire), tapez AT+BAUDx, le module devrait répondre OKvitesse (la vitesse qui convient le x entré dans la commande).
- (Note : 1 pour 1200, 2 pour 2400, 3 pour 4800, 4 pour 9600, 5 pour 19200, 6 pour 38400, 7 pour 57600, 8 pour 115200).

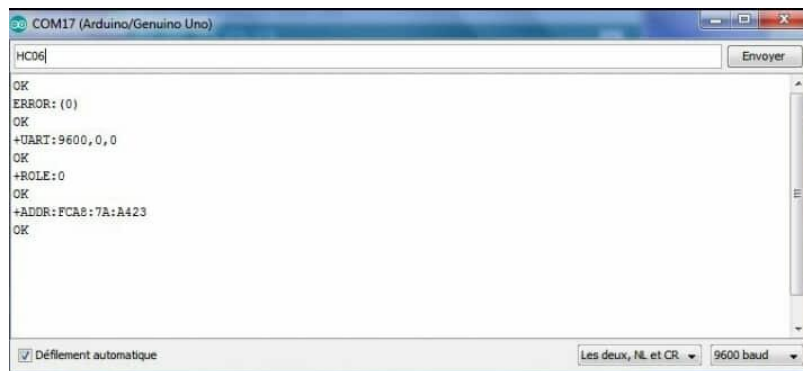


Figure 18

- Nous nous assurons que les mots de passe sur le HC-05 et le HC-06 sont les mêmes
 - Nous trouvons l'adresse du HC-06
 - Nous pouvons le faire avec un appareil Android ou avec le HC-05 comme suite.
- La commande AT+BIND=adresse,du,slave (on a remplacé les : par ,) cela nous permet de connecter un module maitre à un esclave.

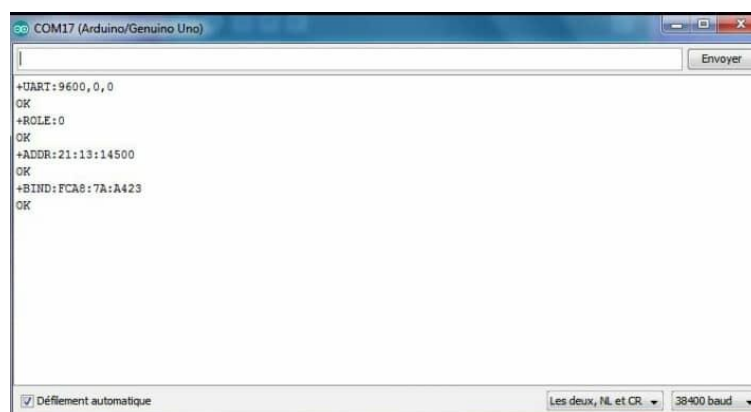


Figure 19

IV.4 Le montage entre deux Arduino-Bluetooth :

Arduino 1, le périphérique maître, ne dispose que du module Bluetooth. Arduino 2, le périphérique esclave, nous avons le module Bluetooth et une LED (avec une résistance appropriée) sur la broche 10

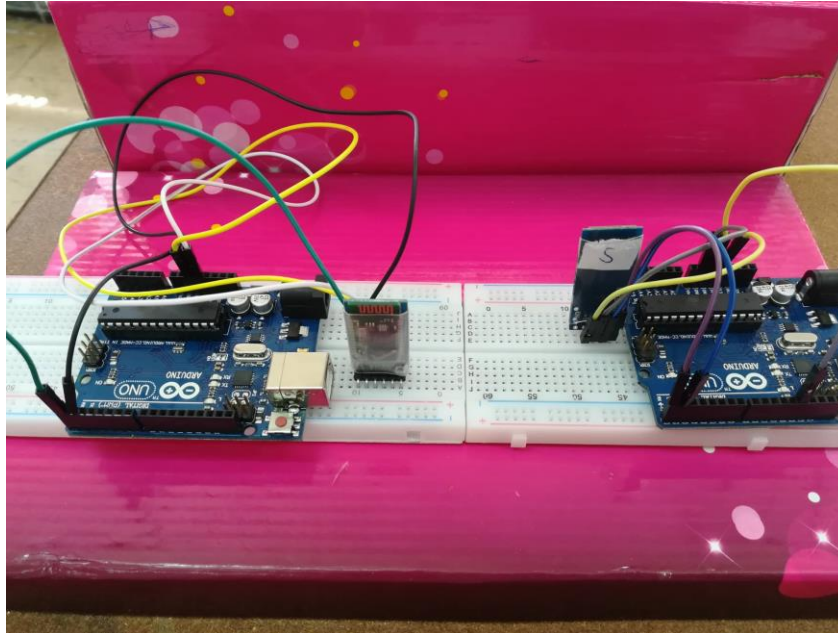


Figure 20

IV.5 Partie programmation :

La carte Arduino uno est une carte programmable, elle peut être programmée avec le logiciel Arduino représenté.

IV.5.1 Le programme Arduino 1 « Maître » :

Le principe consiste à remarquer que dans Arduino 1, le périphérique maître connecté au HC-05 envoie simplement la commande LEDON, attend une seconde, envoie les commandes LEDOFF attend une seconde, puis se répète indéfiniment.

```
#include <AltSoftSerial.h>
AltSoftSerial BTserial;
// Change DEBUG to true to output debug information to the serial monitor
boolean DEBUG = true;
void setup()
{
  if (DEBUG)
  {
    // open serial communication for debugging and show
    // the sketch filename and the date compiled
    Serial.begin(9600);
    Serial.println(__FILE__);
    Serial.println(__DATE__);
  }
}
```

```

    Serial.println(" ");
  }
  // open software serial connection to the Bluetooth module.
  BTserial.begin(9600);
  if (DEBUG) { Serial.println("BTserial started at 9600"); }
} // void setup()

void loop()
{
  BTserial.println("<LEDON>");
  if (DEBUG) {Serial.println("LEDON command sent");}
  delay (1000);
  BTserial.println("<LEDOFF>");
  if (DEBUG) {Serial.println("LEDOFF command sent");}
  delay (1000);
}

```

On remarque que les commandes sont contenues dans les marqueurs de début et de fin, <et>.

```

BTserial.println("<LEDON>");
BTserial.println("<LEDOFF>");

```

L'utilisation des marqueurs de début et de fin permet au périphérique destinataire de vérifier qu'il dispose d'une commande complète avant d'agir.

IV.5.3 Le programme Arduino 2 « esclave » :

Dans Arduino 2, le périphérique esclave vérifie la présence de données et s'il existe un marqueur de départ, il commence à mettre les données reçues dans la variable receiveChars []. Lorsqu'un marqueur de fin est reçu, il attribue la valeur TRUE à l'indicateur newData. Toute donnée ne figurant pas entre les marqueurs de début et de fin est ignorée.

```

#include <AltSoftSerial.h>
AltSoftSerial BTserial;

// Change DEBUG to true to output debug information to the serial monitor
boolean DEBUG = true;
// Variables used for incoming data

const byte maxDataLength = 20; // maxDataLength is the maximum length allowed
for received data.
char receivedChars[maxDataLength+1] ;
boolean newData = false; // newData is used to determine if there is a new
command

```



```

void setup()
{
  // LED on pin 3
  pinMode(3, OUTPUT);
  digitalWrite(3,LOW);

  if (DEBUG)
  {
    // open serial communication for debugging and show the sketch name and the date
compiled
    Serial.begin(9600);
    Serial.println(__FILE__);
    Serial.println(__DATE__);
    Serial.println(" ");
  }
  // open software serial connection to the Bluetooth module.
  BTserial.begin(9600);
  if (DEBUG) { Serial.println(F("AltSoftSerial started at 9600")); }
  newData = false;
} // void setup()

void loop()
{
  recvWithStartEndMarkers();          // check to see if we have received any new
commands
  if (newData) { processCommand(); } // if we have a new command do something
}
void processCommand()
{
  newData = false;
  if (DEBUG) { Serial.print("recieved data = "); Serial.println(receivedChars); }
  if (strcmp ("LEDON",receivedChars) == 0) { digitalWrite(13,HIGH); }
  else if (strcmp ("LEDOFF",receivedChars) == 0) { digitalWrite(13,LOW); }
// function recvWithStartEndMarkers by Robin2 of the Arduino forums
// See http://forum.arduino.cc/index.php?topic=288234.0
void recvWithStartEndMarkers()
{
  static boolean recvInProgress = false;
  static byte ndx = 0;
  char startMarker = '<';
  char endMarker = '>';
  if (BTserial.available() > 0)
  {
    char rc = BTserial.read();
    if (recvInProgress == true)
    {
      if (rc != endMarker)
      {
        if (ndx < maxDataLength) { receivedChars[ndx] = rc; ndx++; }

```

```

    }
    else
    {
        receivedChars[ndx] = '\0'; // terminate the string
        recvInProgress = false;
        ndx = 0;
        newData = true;
    }
}
else if (rc == startMarker) { recvInProgress = true; }
}
}

```

Lorsque newData est TRUE, nous savons que nous avons une commande à traiter. Dans ce cas, nous configurons la broche 10 HIGH ou LOW pour allumer ou éteindre la LED.

```

if (strcmp ("LEDON",receivedChars) == 0) { digitalWrite(3,HIGH); }
else if (strcmp ("LEDOFF",receivedChars) == 0) { digitalWrite(3,LOW); }

```

Test de connexion :

La LED du HC-05 doit avoir 2 clignotements rapides toutes les 2 secondes (ou plus).

Le voyant sur le HC-06 doit être allumé (ne clignote pas).

Nous avons maintenant connecté le HC-05 et le HC-06, le HC-05 se connecte automatiquement au HC-06 et ce dernier sera allumé.

Une fois la connexion établie, le HC-05 passe en mode de communication. Si on veut continuer avec le mode AT, on doit réinitialiser le module avec la broche 34 HIGH ou avec le commutateur à bouton fermé.

On réinitialise les deux Arduinos et on coupe l'alimentation du HC-05 et du HC-06.

Nous allumons le HC-06, le voyant doit clignoter rapidement 5 fois par seconde. Cela indique qu'il attend une connexion ou qu'il doit être jumelé.

Nous allumons le HC-05, la LED clignotera plusieurs fois lors de la première utilisation, puis changera régulièrement en clignotant toutes les deux secondes.

Le voyant sur le HC-06 s'allumera à plein temps (aucun clignotement), les modules sont maintenant connectés.

On ouvre les moniteurs en série et tout ce que vous entrez dans l'un du module apparaîtra dans l'autre.

Chaque fois que les modules sont sous tension, ils doivent se connecter. Si on constate qu'ils ne se connectent pas, on fait rallumer les Arduinos.

Parfois, le HC-05 reste bloqué pour établir une connexion, à cette situation, le voyant du HC-05 sera soit :

1. Clignote rapidement 3 fois toutes les deux secondes.
2. Clignote deux fois par seconde.

Cela se produit normalement lorsque le HC-06 est éteint puis rallumé sans réinitialiser le HC-05. Les périphériques semblent se connecter plus rapidement au premier démarrage du HC-06.

V. CONCLUSION :

Le projet qu'on a réalisé c'est la communication entre deux Arduinos par Bluetooth ,dans cela on conclue :

la communication Bluetooth est très utile et suffisante pour échanger des données entre deux Arduinos .

Ce projet s'est révélé très enrichissant dans la communication ou il a consisté en une approche concrète avec l'utilisation des composants électronique. on pourrait penser a des améliorations pour ce projet comme ajouter des composants plus complexes et très avancés tel que un capteur de température et un capteur d'humidité et de les contrôlé